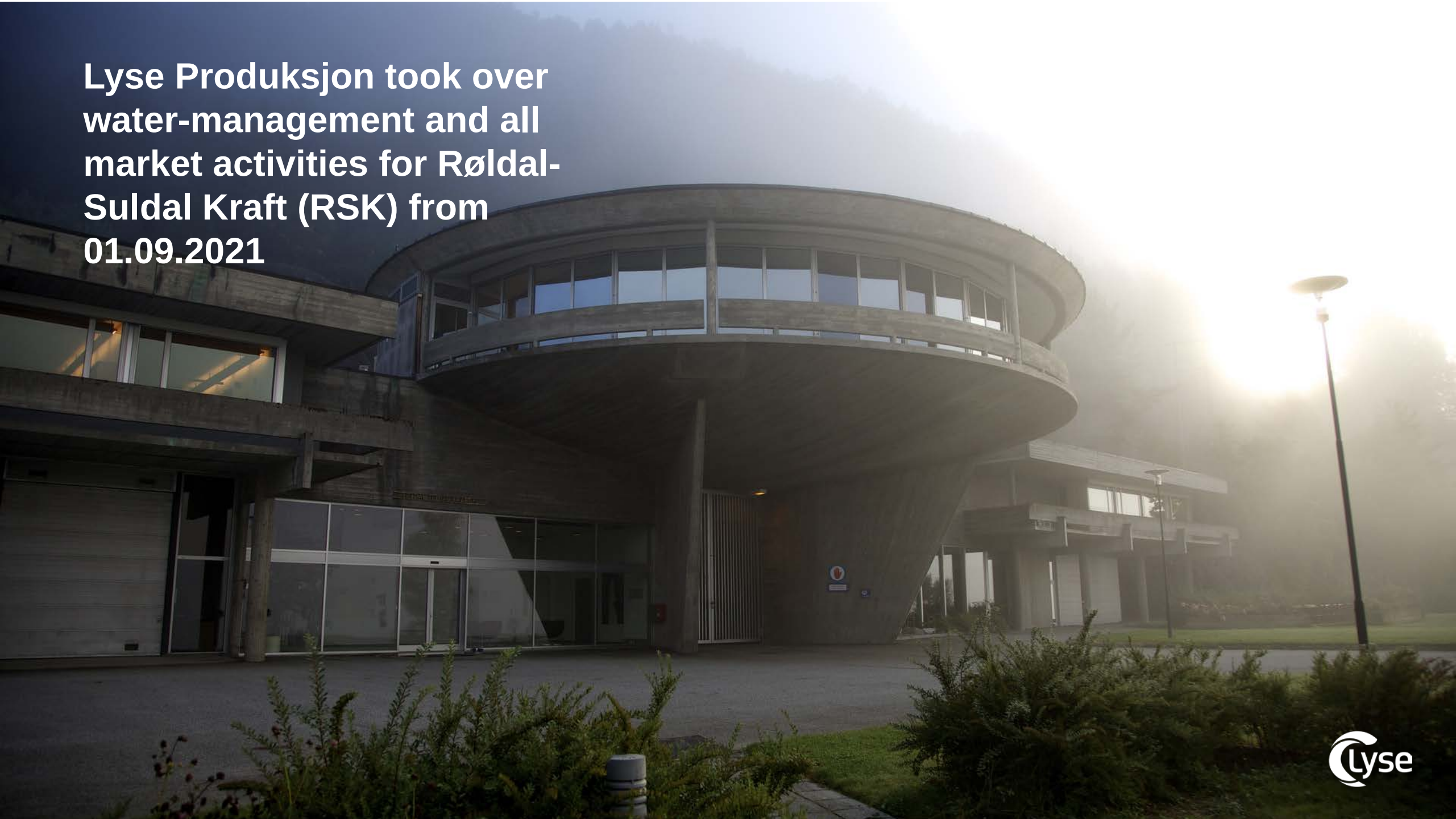
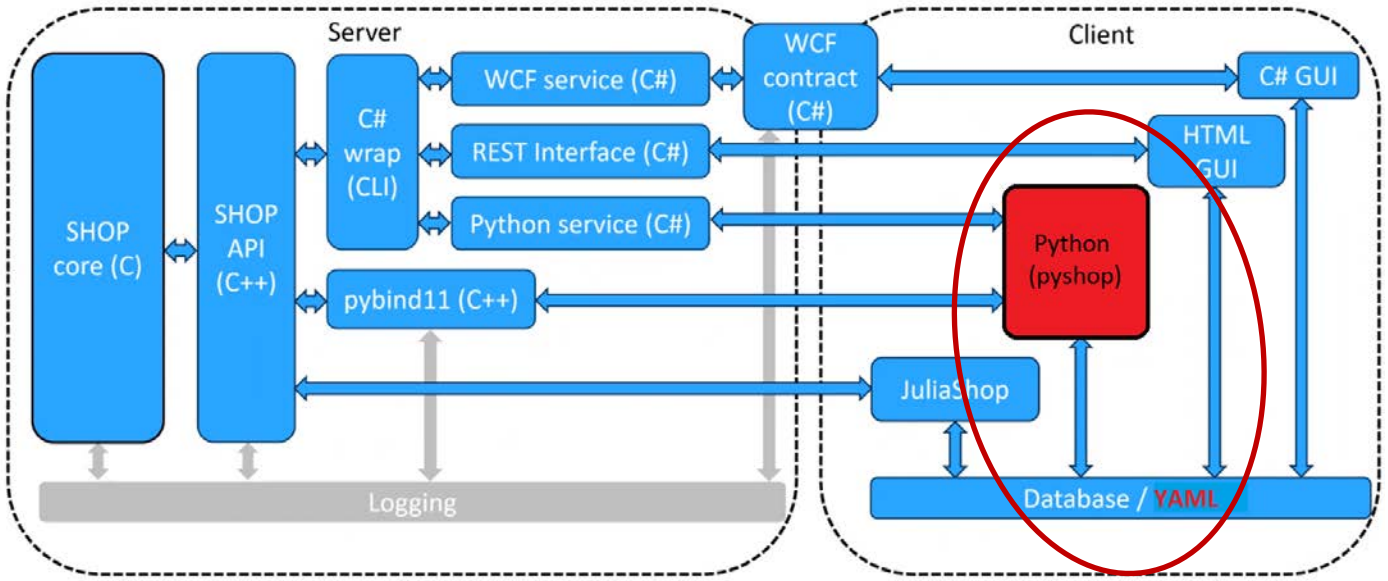
SHOP experiences

Lyse Produksjon AS, User forum and -meeting Hydro Scheduling 2021

Lyse Produksjon took over water-management and all market activities for Røldal-Suldal Kraft (RSK) from 01.09.2021



We went for pySHOP with model description in YAML



SHOP YAML case

SHOP YAML case consist of four different types of content.

YAML case :

- time
- model
- connections
- commands

model example

The following example is based on a real SHOP YAML case where some attributes a objects have been omitted to give a better overview:

```
model:
  reservoir:
    Reservoir1:
      lrl: 90
      hrl: 100
      vol_head:
        ref: 0
        x:
          - 0
          - 12
          - 14
        y:
          - 90
          - 100
          - 101
```

Project proposal

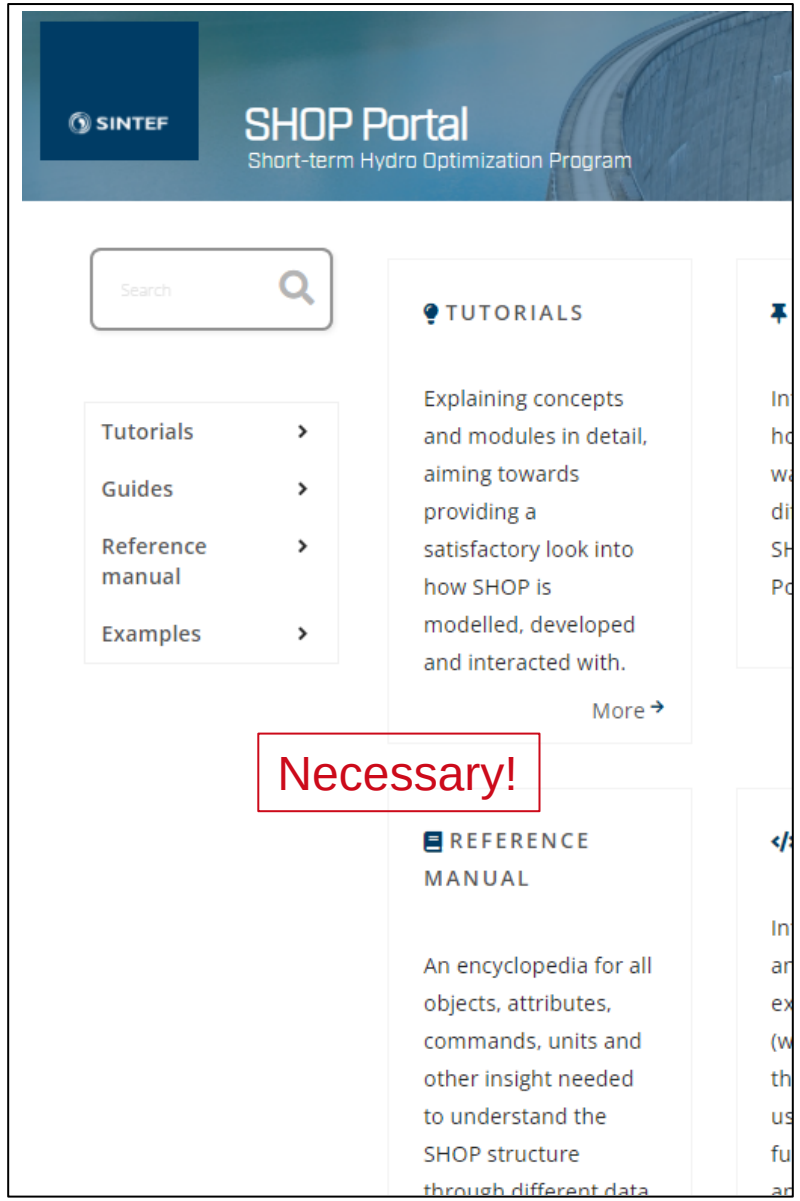
SHOP Support Lyse

VERSION
1

PROSPECTIVE CLIENT
Lyse Produksjon As

PROJECT NO.

Not that necessary!
(or at least not as necessary as we thought)



Necessary!

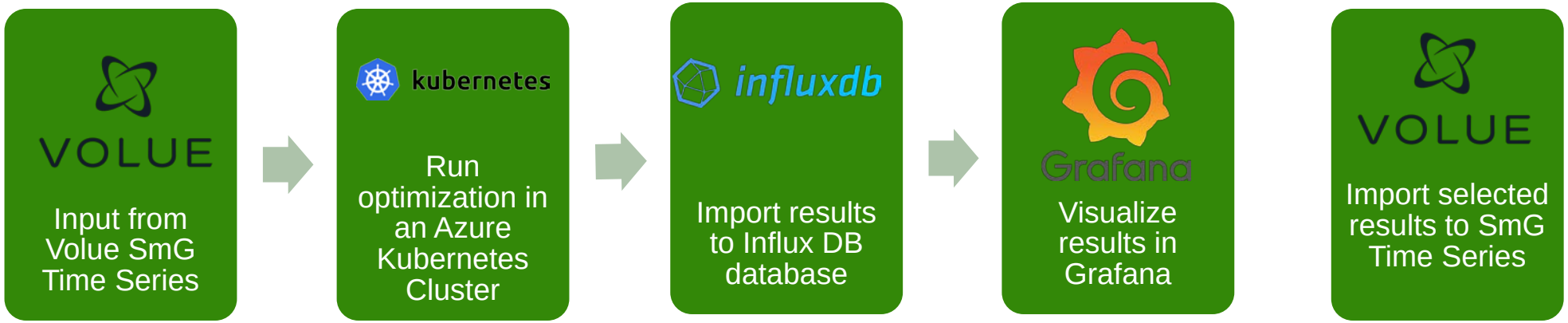
```
ss ShopCaseBaseClass:

def __init__(self, source, shop_init_func=pyshop.ShopSession):

    log.info(f'Init ShopCase : source_type={type(source)}')

    self.shop_init_func = shop_init_func
    self.case = None

    if isinstance(source, pyshop.ShopSession):
        self._from_shopsession(source)
    elif isinstance(source, (dict, DictImitator)):
        self.case = source
    elif isinstance(source, bytes):
        self._from_bytestring(source)
    elif isinstance(source, str) and source[0] == '{':
        self._from_json(source)
    #elif isinstance(source, 'yaml_string'):
    #    raise NotImplementedError
    elif isinstance(source, (str, Path)) and (Path(source) / 'model.yaml').exists()
        self._from_dir(source)
    elif isinstance(source, (str, Path)) and '.shop.zip' in Path(source).name:
        self._from_file(source)

    if isinstance(self.case, dict):
        self.case = DictImitator(**self.case)

@property
def model(self):
    return self.case['model']

@property
def time(self):
    return self.case['time']

@property
def commands(self):
    return self.case['commands']

@property
def connections(self):
    return self.case['connections']
```

Really helpful!

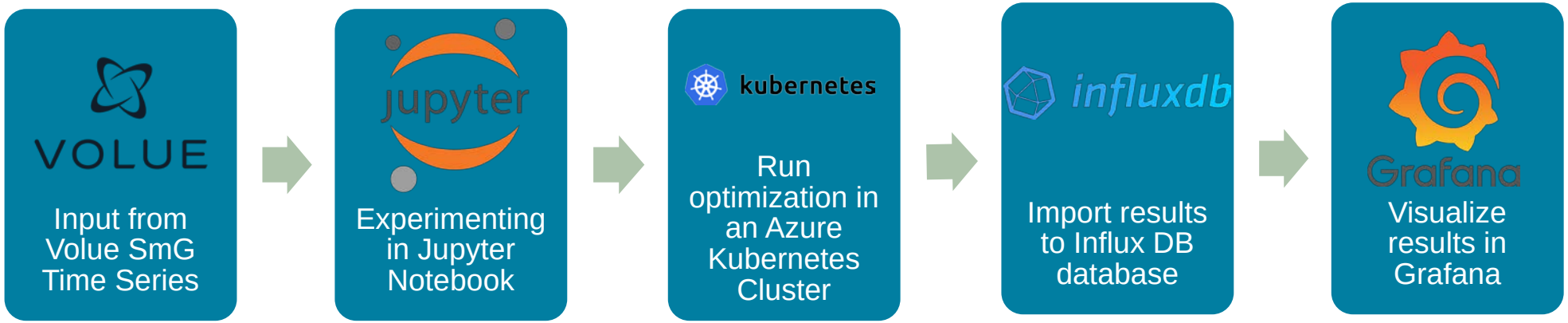
https://github.com/nilsra/shopcase

Our architecture

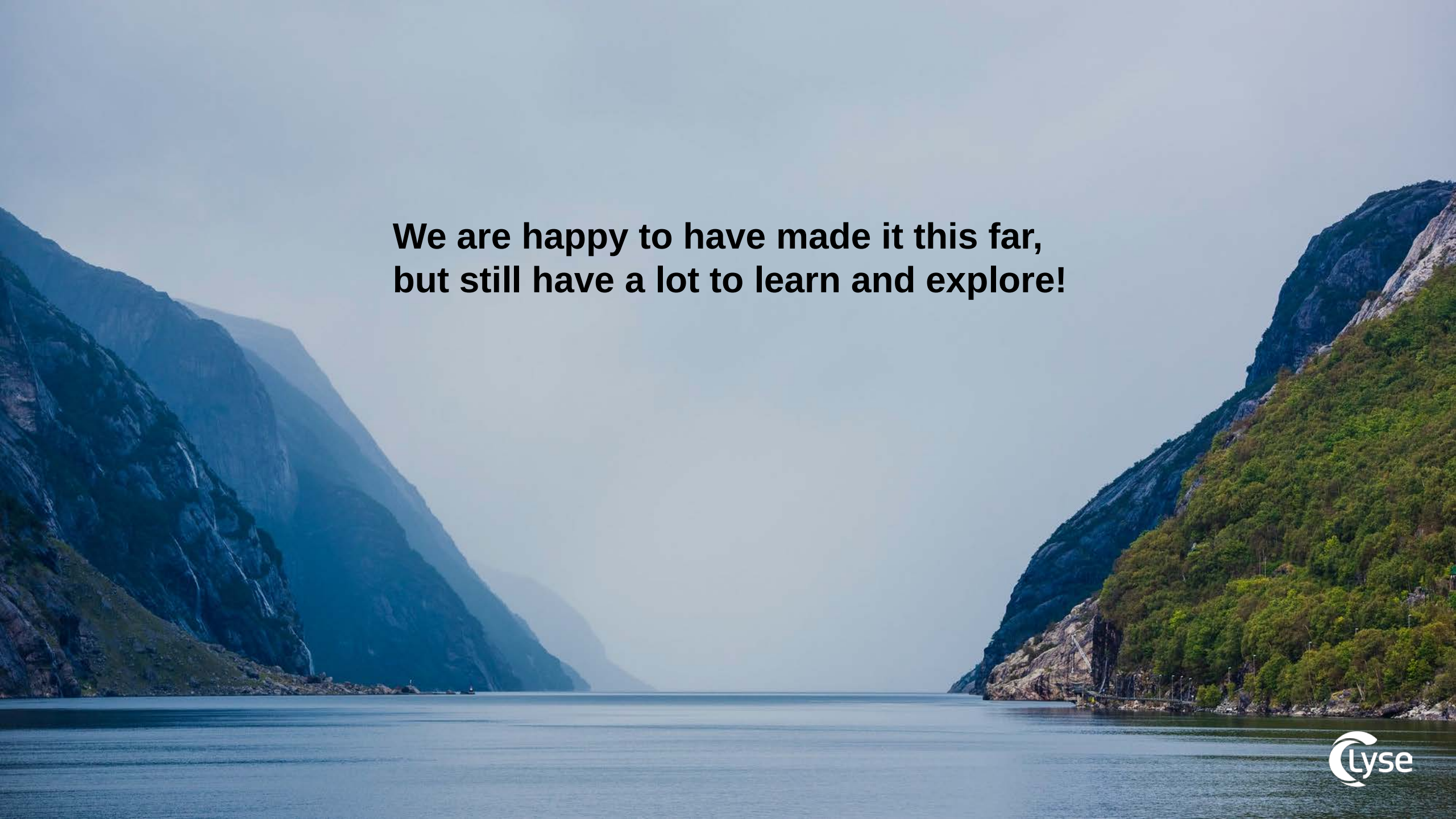
Operational:



Experimental:



lyse


We are happy to have made it this far,
but still have a lot to learn and explore!


Lyse